Efficient Management of Transportation Logistics related to Animal Disease Outbreaks

H. Vernieuwe*[,a], E. Ducheyne[b], G. Hendrickx[b], B. De Baets[a]

[a]*Ghent University, Department of Applied Mathematics, Biometrics and Process Control, Coupure links 653, 9000 Gent, Belgium*
[b]*Avia-GIS, Risschotlei 33, 2980 Zoersel, Belgium*

Abstract

Vehicle routing is a key instrument to manage and control animal disease outbreaks. This paper focuses on an efficient, user-friendly and automatic procedure to manage transportation logistics to and between farms in the case of an outbreak. This procedure can be embedded into a veterinary geographical information system for the management and control of disease outbreaks. The transportation logistics for the problem at hand can be divided into two main transportation categories: (i) round itineraries, which are special cases of the travelling salesman problem, and (ii) one-to-one itineraries. Attention is given to the use of user-friendly, heuristic yet efficient algorithms for the determination of these itineraries. It is furthermore shown that the procedure is developed in such a way that the identified routes meet both national and international regulations in force during disease outbreaks.

Key words: disease outbreak management, transportation management, veterinary disease information systems

1. Introduction

Both contagious as well as non-contagious vector-borne diseases can lead to enormous economic losses, see for instance the 1997–1998 outbreak of Classical Swine Fever in The Netherlands [Meuwissen et al., 1999] and the 2003 outbreak of Foot-and-Mouth Disease in the UK [Kao, 2003]. Furthermore, zoonoses such as avian influenza pose an additional threat to the human population. Since a

*Corresponding author.
Email addresses: `Hilde.Vernieuwe@UGent.be` (H. Vernieuwe), `educheyne@aviagis.be` (E. Ducheyne), `ghendrickx@aviagis.be` (G. Hendrickx), `Bernard.DeBaets@UGent.be` (B. De Baets)

timely response during the first stage of an outbreak can limit disease spread, efficient management of animal disease outbreaks is important.

At the national level, different software packages are available to collect, store and analyse data. Packages have been developed by international organizations (e.g. the Transboundary Animal Disease Information System, TADinfo, for EMPRES-i), national organizations (e.g. Center for Epidemiology and Animal Health, USA), research groups (EpiMAN by EpiCentre Massey University, New Zealand) and private companies (e.g. Vet-geoTools currently being developed by Avia-GIS, Belgium). The integration of field disease data, environmental data and remotely sensed derived products within a veterinary Geographical Information System (GIS) contributes to the understanding of the disease epidemiology during peace time, and when applied during a state of crisis, helps to manage the outbreak more rapidly ([Hendrickx et al., 2004], [Rizzoli et al., 2004], [Conte et al., 2005], [Cringoli et al., 2005], [de La Rocque et al., 2005], [Kroschewski et al., 2006], and [Pinzon et al., 2005]). However, veterinary GISs are rarely used in operational decision making [Hendrickx et al., 2004].

An important task for the government during a disease outbreak is to eliminate possible disease transmission by contaminated vehicles. Following official regulations, a quarantine and surveillance zone are usually delineated around infected farms, within which specific sanitary measures are imposed. Some vehicle activities to and between farms located within these zones may continue if they adhere to strict rules. These rules could be to avoid trespassing a surveillance zone or to go to disinfection points prior to leaving a quarantine zone. Examples of vehicle routes include veterinary farm visits, milk collection rounds, collection of cadavers, *etc.*

Nowadays, scheduling of the routes is mostly set up by hand following a predetermined scenario, which is very time-consuming. In addition, the schedules may suffer from unavoidable human weaknesses and may therefore be suboptimal. Hence, this paper focuses on the use of an automated procedure, which identifies minimal cost vehicle routes that try to avoid a potential disease spread. By integrating this scheduling into Vet-geoTools, which can access frequently changing field data, disease outbreaks can be managed more efficiently and rapidly. Two major vehicle routing types can be distinguished: round itineraries, whether or not capacitated, that visit several farms in one round and one-to-one visits that collect goods at a particular farm and directly deliver these at a depot.

This paper is organized as follows. Section 2 divides the scheduling problem into two subproblems for which suitable existing algorithms are identified and described in the corresponding subsections. Section 3 describes the specific precedence constraints and the identification of the schedules for round transports, in particular the non-capacitated veterinary visits and the capacitated milk collection rounds, whereas Section 4 describes the needs and the identification for the one-to-one transports, in particular the collection of cadavers. These schedule identification tasks were performed on the basis of an existing road map and a real-life scenario of a historical disease outbreak for which the quarantine, surveillance and free zones were delineated.

2. Suitable algorithms for transportation management

In essence, the problem of identifying a feasible schedule or route for each of the above-mentioned types of vehicle movement depends upon two subproblems: first, a route of minimal risk needs to be identified between two possibly subsequent locations in the tour, and second, based on these routes, a feasible schedule needs to be identified.

2.1. One-to-one minimum path finding problems

The first subproblem can be considered as belonging to the group of one-to-one minimum path finding problems. An overview of heuristic algorithms for this type of problems is given by Fu et al. [2006]. According to them, the A* algorithm [Hart et al., 1968] is the most popular among all heuristic algorithms and saves 50 % in computation time as compared to an ordinary Label-Setting algorithm such as the algorithm of Dijkstra [1959]. Furthermore, several other ideas, such as the use of a bi-directional search or a hierarchical search, have been proposed in order to increase the computational efficiency of path finding algorithms. However, in the case of the bi-directional search, the total computational efficiency is limited for transportation networks [Fu et al., 2006]. Conversely, the hierarchical search's savings in computation time could be of several orders of magnitude [Fu et al., 2006]. Nevertheless, its implementation is more complex due to the fact that a hierarchical road network consisting of an undetermined number of layers has to be identified out of a real road network and the search transition between the hierarchial layers needs to be controlled [Car and Frank, 1994]. Therefore, given the specific properties inherent to the two different types of transportation, the A* algorithm (see Section 2.3) was selected in order to find the one-to-one minimum path for the capacity and veterinary-related transportation problems. The A* algorithm was used hierarchically based on a two-level road network, *i.e.* one level for the main roads, highways, *etc.*, and a second level for the smaller roads, in order to determine the route for transportation of for instance cadavers.

2.2. The travelling salesman problem

The second subproblem belongs to the group of travelling salesman problems, which can be easily formulated but are difficult to solve. Suppose a salesman has to visit N predefined cities in order to sell his products, the problem is then to identify the shortest tour that visits all cities exactly once whilst starting and ending in the same city. As shown by Garey and Johnson [1979], this problem is NP hard and one of the most important test cases for new combinatorial optimisation algorithms. The problem at hand can furthermore be regarded as a special instance of the travelling salesman problem: precedence constraints are supplied and hence they can be classified among the sequential ordering problems (SOP) or precedence-constrained travelling salesman problems. Several heuristic algorithms have already been employed in order to find the best possible route for the travelling salesman problem and its variants. Pisinger and Ropke [2007] and Ropke and Pisinger [2006] used an adaptive large neighbourhood search as local search method embedded in a main model based on simulated annealing. Bianchessi and Righini [2007] applied tabu search combined with a local search heuristic for simultaneous pickup and delivery problems. Tavakkoli-Moghaddam et al. [2006] and Tavakkoli-Moghaddam et al. [2007] used a hybrid model based on simulated annealing and a 1-opt and 2-opt-based neighbourhood search. Ganesh and Narendran [2005] developed a heuristic based on clustering and genetic algorithms (CLOSE) to solve asymmetric precedence-constrained travelling salesman problem. Genetic algorithms have also been employed in order to search for an optimal, least cost solution for the collection of milk from farms [Dooley et al., 2005]. Pacheco and Martí [2006] employed tabu search and different constructive solution methods for a multi-objective routing problem. In order to avoid parameter tuning and modifications, which is a drawback of the majority of the heuristic algorithms, Nikolakopoulos and Sarimveis [2007] introduced a new heuristic algorithm, *Threshold Accepting* (TA), an algorithm similar to simulated annealing, combined with an intense local search in order to find an optimal solution for three special instances of the travelling salesman problem, among which is included the sequential ordering problem. Their algorithm has been tested on a variety of artificial and real life problems and its computational efficiency has been demonstrated. Furthermore, good qualitative results were obtained. In order to schedule the transportation of live animals following veterinary rules, Sigurd et al. [2004] reported the use of dynamic programming. As the scope of this research is to manage the transportation logistics in zones of disease outbreaks as efficiently as possible, user-friendliness, robustness and efficiency of the algorithm were important criteria. Therefore, preference was given to the algorithm of Nikolakopoulos and Sarimveis [2007] (see Section 2.4).

2.3. Identification of the path between two nodes

The shortest path for the transportation problems between the veterinarian's practice or the milk factory and the farms to be visited and the farms in between is determined based on a graph $G = (N, E, W)$ with N the set of nodes. The set of nodes is composed of the location of veterinarians' practices, milk factories, the farms to be visited and the road crossings, and each node has a corresponding risk level associated with the zone, *i.e.* quarantine, surveillance or free, it is situated in. The set E contains the edges between the different nodes and has a distance and maximum allowed velocity associated to it defined as a weight $w \in W$. The A* algorithm starts from the start node and calculates for every adjacent node n_i a cost:

$$F_i = L_i + a_{i,d} \, , \tag{1}$$

with L_i the cost to travel from the start node n_o to node n_i and $a_{i,d}$ the heuristic value of the estimated travel cost from node n_i to the destination node n_d. In a next step, the node n_j with minimal F is selected as the next node along the path. The algorithm then continues by calculating F for every adjacent node to n_j, and selecting the node with minimal F out of all already visited nodes which do not take part in the path and so on. The algorithm stops when n_d is reached or if all possible nodes have been visited.

2.4. Identification of the schedule

A feasible schedule for the veterinary and milk transportation can be identified based on a directed graph $G' = (N', E', W')$. N' is the set of nodes, with associated risk level p, which contains the veterinarian's practice or the milk factory and the farms to visit. The set E' contains the edges for which the weights w', expressed in time length (h), were calculated by means of the A* algorithm. The problem can be formulated as the minimisation of the following objective function:

$$\sum_{i \in N'} \sum_{j \in N'} w'_{ij} b_{ij} \, , \tag{2}$$

with $b_{ij} \in \{0, 1\}$, and $b_{ij} = 1$ if one travels from node n'_i to node n'_j with the following constraints:

$$\sum_{j \in N'} x_{ij} = 1 \quad , \forall i \in N' \, , \tag{3}$$

$$\sum_{i \in N'} x_{ih} - \sum_{j \in N'} x_{hj} = 0 \quad , \forall h \in N' \, , \tag{4}$$

$$p_i \geq p_j\,. \tag{5}$$

Condition (3) states that every farms needs to be visited exactly once, condition (4) enforces the transportation to arrive in node h and to leave from node h and condition (5) stipulates that farms located in high risk zones have a higher priority for visiting. This last condition is to be reversed in the case of milk transports. As already quoted, the algorithm of Nikolakopoulos and Sarimveis [Nikolakopoulos and Sarimveis, 2007] is used to calculate an optimal feasible schedule. The basic idea of this algorithm is similar to that of simulated annealing [Kirkpatrick et al., 1983]. The algorithm starts with a randomly selected solution $\mathbf{x}_c \in X$, with X the set of all possible permutations of nodes, for which a feasible schedule $\mathbf{S}_c$ w.r.t. the conditions is identified. Following the order of nodes in $\mathbf{x}_c$, each node is inserted into $\mathbf{S}_c$ into the lowest cost feasible position (see [Nikolakopoulos and Sarimveis, 2007]). Based on one of six predefined local search operators, a neighbouring solution $\mathbf{x}_n$ is identified from $\mathbf{x}_c$, and a feasible sequence $\mathbf{S}_n$ is further identified. The value of the objective function (2) is calculated and compared to the value for $\mathbf{S}_c$:

$$\Delta f = f(\mathbf{S}_n) - f(\mathbf{S}_c)\,. \tag{6}$$

If $\Delta f \leq T$ with T an adaptive threshold value, $\mathbf{S}_n$ is accepted as the new schedule. It is important to note that values of T different from zero enable the algorithm to escape from local optima in order to be able to achieve better solutions. A sorted set of possible threshold values TS is used and automatically adjusted during the execution of the algorithm. Eventually, TS will only contain elements equal to zero. In reality, however, it is impractical to let all elements of TS become zero. Therefore, a maximum number of iterations is identified as to shorten the CPU usage of the algorithm.

A second issue in the identification of a feasible schedule is the fact that a schedule might be divided into shorter trips if a maximum duration and/or a maximum load capacity of the transportation vehicle is exceeded. In order to address this, the *Split* algorithm, introduced by Prins [2004] in the framework of an evolutionary optimisation algorithm, is employed. *Split* optimally divides a schedule S into several shorter trips given a predefined maximum duration and/or maximum capacity and acts on a graph $G'' = (N'', E'', W'')$, with N'' the set of nodes, E'' the set of edges e_{ij} from nodes n_i to n_j if travelling from n_i to n_j is allowed given the travel cost and capacity, and W'' the set of weights w_{ij} equal to the travel costs from n_i to n_j. Furthermore the costs and capacities to pick up or deliver at the nodes are taken into account. In order to use the cost of the possibly divided schedule, *Split* is embedded into *Threshold Accepting*.

3. The identification of schedules for round transports

3.1. Veterinary schedules

In case of disease outbreaks, a veterinarian is obliged to visit all farms assigned to his practice for sampling. Farms situated in quarantine zones hereby take priority over farms situated in surveillance zones. The remaining farms, *i.e.* farms located in the free zone, will be visited last. Afterwards, all farms are inspected weekly. Furthermore, the veterinarian is encouraged to avoid quarantine and/or surveillance zones unless the destination is located inside those zones. A last condition enforces that whenever the veterinarian leaves a quarantine zone, his vehicle needs to be disinfected. In order to determine the schedule for the veterinarian, the A* algorithm is initially used in order to identify the one-to-one paths, *i.e.* the paths from the veterinarian's practice to the farms and vice versa and the paths between the farms. Based on the risk levels associated with the farms, some one-to-one paths are not allowed and will therefore not be determined. If farm A is located for instance in a surveillance zone and farm B in a quarantine zone, then it is clear that given the precedence constraints, the veterinarian is not allowed to travel from A to B. Based on these predefined paths, the *Threshold Accepting* algorithm and the *Split* algorithm were used to determine the final schedule possibly divided into trips.

3.1.1. Identification of the one-to-one shortest paths

As already mentioned (see Section 2.3), the A* algorithm uses the travel cost along the current path and a heuristic value to estimate the travel cost from the current node to the destination node to find the shortest path. If travel cost is expressed using distance units, the Euclidean distance is most commonly used as heuristic, since the algorithm needs a lower limit to ensure the shortest path is found. Similarly, travel cost can be expressed using time units, in which case the fastest route is sought. In order to define the lower limit of the remaining travel time, the Euclidean distance is calculated and converted into a corresponding time-based heuristic using the maximum allowed velocity found in the road network. In order to restrict the risk of spread of diseases, the following boundaries were supplied:

1. If the current node is located in a zone of higher risk than the preceding node on the path, a penalty distance (time) is added to the current distance (travel time) in order to discourage the traversing of zones of higher risk;
2. Whenever the veterinarian leaves a quarantine zone, *i.e.* the preceding node along the path is located in a quarantine zone whereas the current node is located in a surveillance zone, a disinfection time or corresponding disinfection distance is added to the current travel cost.

3.1.2. Identification of the schedule

In order to determine the schedule for the veterinarian, the travel distances or times, calculated by means of the A* algorithm are first converted, if necessary, into travel times (h) and stored into a weight matrix that serves as a basis for *Threshold Accepting* and *Split*. As already mentioned, *Threshold Accepting* requires a single parameter, *i.e.* the maximum number of iterations. In order to determine this parameter, the maximum number of iterations was altered from $100, 200, \ldots, 1000$ with 30 repetitions for 23 test cases with visits ranging from 2 to 55 farms (see Table 1) identified on the basis of a real-life data set. It was furthermore assumed that trips have a maximum duration of 10 h and that a farm visit lasts 4.5 h. The calculation of the one-to-one paths was performed distance-based and a penalty of 10 km was added for entering a zone of higher risk. A disinfection time of 0.5 h, converted to a corresponding distance of 25 km, was assumed. Figure 1 shows the minimal and maximal costs out of 30 repetitions for a different maximum number of iterations for a veterinarian who has to visit 55 farms. This figure shows that several costs can be found, indicating that suboptimal schedules are identified. Figure 1 (a) furthermore reveals that the difference between the worst and the best schedules found is at most 0.1 h. For the majority of the veterinarians, however, only a single cost is found irrespective of the maximum number of iterations from which can be concluded that schedules close to optimality will be identified if a maximum number of iterations equal to 100 is used. Therefore 100 iterations were used throughout the rest of the study.

Influence of the penalties for entry in zones of higher risk. In case of disease outbreaks, quarantine and surveillance zones are delineated around an infected farm. To discourage entry of these zones, penalties are added to paths that cross them during the path search. However, these penalties cannot be set too high as these zones need to be entered or traversed in some particular cases, e.g. if the farm to visit is located in a quarantine or surveillance zone, or if the only possible way to a farm runs through them. Furthermore, in case of a very high penalty, the path-finding algorithm will initially search for paths with a length lower than the penalty which may result in a high CPU time. Given these considerations, a first choice of penalty can be half the circumference of the respective zones. Table 2 lists the radii of the zones as imposed for classical swine fever and the penalties (distance- and time-based) that were used throughout this paper for these radii. Figure 2 shows the tour for a veterinarian when no penalty is added versus the tour when a penalty is added for entering a quarantine zone. These figures clearly show that the veterinarian's route trespasses the quarantine zone in order to visit a farm situated on the other side of the zone if no penalty is applied. Conversely, an alternative route that avoids the quarantine zone is identified if a penalty is added to paths that cross it.

Influence of the disinfection locations. In case of disease outbreaks, vehicles may have to be disinfected if they leave the quarantine zones. In practice, disinfection locations are either established at fixed locations at the border of the quarantine zone or are mobile stations with a changing location during the epidemic. In the first case, nodes with a disinfection attribute receive a code that the quarantine zone is accessible. If the A* algorithm tries to identify a path that enters the quarantine zone through a node that has no disinfection attribute, a very high penalty is added to the current cost. If mobile disinfection equipment is used, it is assumed that disinfection always occurs whenever the vehicle leaves the quarantine zone and therefore no penalty is added. Figure 3 shows the difference in route for a veterinarian if a fixed (a) and mobile (b) disinfection unit is assumed. In case of fixed disinfection units the route is changed so that it passes through the indicated location.

3.2. Capacitated transports

When quarantine and surveillance zones are delineated, factories may still collect the milk from dairy farms if dairy cattle is not the susceptible population. However, certain restrictions, similar to those of veterinary visits, are imposed. The factory first collects milk from dairy farms located outside the surveillance and quarantine zones. Farms located in surveillance zones then take priority over farms in quarantine zones, which are visited last. This implies that condition (5) is changed to:

$$p_i \leq p_j \,, \tag{7}$$

for travelling from node n'_i to node n'_j. Furthermore, the transportation is discouraged to trespass surveillance and quarantine zones without reason. If the vehicle leaves a quarantine zone, as is the case for the veterinary visits, a disinfection takes place for which a disinfection time or distance is charged. Similar to the veterinary vehicle routing problem, the A* algorithm is used to determine the one-to-one paths that respect the order given the risk level of start and destination node. The approach for assigning penalties is identical as for the veterinary visits. Afterwards, the *Threshold Accepting* and *Split* algorithms are used to determine a feasible, final schedule possibly divided into trips that respect the maximum capacity of the vehicle and maximum duration of a trip. The identification of the schedule can be performed twofold. First, an already existing schedule optimised for maximum capacity and duration can be adapted in order to account for the precedence constraints. In this case, the separate existing trips are reordered such that milk is collected from farms obeying condition (5). In this case, the possible extra duration of the trip is of no importance to the factory. Second, new trips are identified given the maximum capacity and duration for the trips, *i.e.* a completely new schedule is determined.

3.2.1. Adaptation of existing trips given precedence constraints

For each existing trip, the A* algorithm was first used to identify the one-to-one shortest paths and penalties as listed in Table 2 were applied for entry in the quarantine and surveillance zones and disinfection. It was furthermore assumed that disinfection locations were indicated in advance (fixed positions). Based on these one-to-one shortest paths, the *Threshold Accepting* algorithm was used to reorder the trip as to minimise its duration. Table 3 shows the original order for the existing trips of a milk factory. The newly assigned order given the precedence constraints is listed in Table 4. From these tables, it is clear that each trip has been adapted separately, without a reorganisation of the schedule itself. Each trip first collects milk from a farm located in the free zone (if present), then continues to collect milk from farms located in the surveillance zones and ultimately collects milk of farms situated in quarantine zones. In case existing trips are adapted, it is important to note that it is possible that trips collect milk from farms situated in the three zones (e.g. Trip 5).

3.2.2. Identification of a new schedule with new trips

In order to identify a new schedule, all farms that are customer of the given milk factory are involved in the re-determination of the trips. The A* algorithm is initially used to identify the one-to-one shortest paths taking into account the aforementioned conditions. It was also assumed that disinfection locations were fixed. The weights of these resulting paths were then stored in a weight matrix used as a basis for the *Threshold Accepting* algorithm. As constraints can be added given a maximum duration and/or load capacity, the *Split* algorithm is also used in order to break the schedule into several shorter trips. Table 5 gives an overview of the schedule divided into trips for which a maximum load capacity of 20000 ℓ was imposed. No condition was set w.r.t. the maximum duration of the trips. The volume that has to be collected from the farm and the zone in which the farms are situated are indicated as well. This table reveals that the maximum capacity of 20000 ℓ per trip has been respected and that the farms situated in the free zone are visited first (trips 1 and 2), followed by the farms located in the surveillance zone (trips 2–5) and ultimately the farms located in the quarantine zone (trips 5–7). In contrast to the method used in Section 3.2.1, trips that collect milk from farms situated in the three existing zones are not present. It should also be noted that the schedule now consists of 7 trips instead of 6, which is due to the fact that the *Split* algorithm tries to break the schedule into trips that fulfill the capacity requirements, yet have the lowest cost possible. If an additional restriction for the maximum trip duration is fixed, *Split* can also be used. Table 6 shows the trip costs of a schedule that has been broken down into trips with a maximum duration of respectively 12 h and 5 h and a maximum load capacity of 20000 ℓ. The load to be collected for each trip is presented as well. Table 6 shows that both requirements have been fulfilled. For the maximum trip duration of 12 h, it can be seen that none of the trips lasts longer than 5.5 h, from which it can be concluded that the maximum capacity was the only restriction used by *Split*. Changing the maximum trip duration to 5 h, one can see that the first trips remain the same in cost and capacity. The other trips have been rearranged as to meet the imposed requirements.

4. One-to-one transportation

4.1. Identification of the shortest paths

With respect to the collection of cadavers and similar transports, the transportation is in essence a one-to-one transportation: cadavers are collected at the farm and directly transported to the destruction company. Therefore, the A* algorithm is used to identify the optimal route that fulfills several subsequent conditions:

- If the transportation leaves the farm, the vehicle is disinfected. However, if this is impossible due to logistic reasons, the closest disinfection location is used. In its trip to the closest disinfection location, passing near non-infected farms is discouraged.
- The transportation then continues to the closest highway or principle road and avoids non-infected farms and the unnecessary entry of quarantine or surveillance zones.
- The transportation then stays as long as possible on the highway or principle roads.
- The route from the highway to the destruction company avoids passing near non-infected farms.

In order to fulfill these requirements, the A* algorithm is used hierarchically, *i.e.* the road network is split up in two layers: a first layer consists of all roads, a second layer consists of main roads and highways only, subroutes are then calculated on the first or second layer, depending on their requirements. If it is not possible to disinfect the vehicle at the farm itself, the closest disinfection location within a distance given by the radius of the quarantine zone is sought for. The first part of the path finding then consists of the identification of the route from the farm to the selected disinfection location. Next, as it is possible that the route to the closest node (in Euclidean distance) on the main road or highway does not correspond to the shortest route, the 20 closest nodes (in

Euclidean distance) from main roads and highways are sought and the A* algorithm is used to identify the route to these selected nodes. The route with the lowest cost is selected as the next part of the route. Subsequently, the 20 Euclidean closest nodes from main roads and highways near the destruction company are identified and the A* algorithm is used in order to determine the route from these selected nodes to the destruction company. Finally, the route along the main roads and highways is identified based on the second layer.

4.2. Influence of the disinfection locations

If the vehicle can be disinfected at the farm itself, the shortest (fastest) route that meets the requirements to the nearest highway or main road is identified. However, if the vehicle cannot be disinfected on the farm, the closest disinfection location is identified, and the shortest (fastest) route to this location is calculated first, subsequently the shortest (fastest) route to the highway or main road is determined. Figure 4 shows the path for a cadaver transportation in case the vehicle is disinfected at the farm (a) or if a disinfection location has to be searched for (b) and also shows that the path follows the main roads (indicated in black) as long as possible.

4.3. Influence presence of non-infected farms

When the routes to the closest disinfection location, the closest main road or highway and the route from the main road or highway to the destruction company are identified, the route should avoid passing non-infected farms. Therefore, similarly as for avoiding unsollicited entry of quarantine and/or surveillance zones, a penalty is added to paths that pass non-infected farms. For the test cases addressed in this paper, a penalty of 10 km was added. As Figure 5 illustrates, the transportation is discouraged to pass near non-infected farms (path indicated in cyan). If for the same transportation, no penalty would be added, the transport follows a path that passes more non-infected farms (path indicated in magenta).

5. Conclusion

As the efficient organisation of transportation logistics in case of disease outbreaks is highly important in order to minimize the spread of disease, this paper focused on the identification of an automatic procedure to organize transportation logistics between farms following specific sanitary regulations in case of disease outbreaks. Two main transportation types could be distinguished: round transports, such as rounds of veterinary farm visits, milk collection, *etc.*, and one-to-one transports, such as the collection of cadavers to a destruction company. This paper showed that by combining the A* algorithm [Hart et al., 1968], the *Threshold Accepting* algorithm [Nikolakopoulos and Sarimveis, 2007] and the *Split* algorithm [Prins, 2004], optimal paths for round transportation, whether or not capacitated and split into shorter trips given trip duration and/or capacity, can be identified automatically taking into account the specific sanitary regulations inherent to the transportation type, such as the use of predetermined disinfection locations, the avoidance of unnecessary trespassing of quarantine and/or surveillance zones, precedence constraints w.r.t. the order in which farms have to be visited and so on. Based on a hierarchical implementation of the A* algorithm, routes that meet the specific rules for one-to-one transports, *i.e.* avoidance of passing near non-infected farms, preference of the use of principal roads, *etc.*, can be identified automatically. However, it should be noted that a maximal benefit can be drawn from this automatic procedure if it takes part in a veterinary GIS system. In this system, a connection with national data bases can be established such that access to frequently changing disease data is assured. Hence, routes can efficiently be calculated at the crisis center by trained operators and handed over to the veterinarians and firms involved.

Acknowledgement

This work has been performed in the framework of the IWT innovation project 060492/AVIA-GIS, financed by the Agency for Innovation by Science

and Technocology, Flanders, Belgium.

References

N. Bianchessi and G. Righini. Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. *Computers and Operations Research*, 34:578–594, 2007.

A. Car and A.U. Frank. General principles of hierarchical spatial reasoning – the case of way-finding. In *Proceedings of the Sixth International Symposium on Spatial Data Handling*, volume 2, pages 646–664, 1994.

A. Conte, P. Colangeli, C. Ippoliti, C. Paladini, M. Ambrosini, L. Savini, F. Dall'Acqua, and P. Calistri. The use of a web-based interactive geographical information system for the surveillance of Bluetongue in Italy. *Revue Scientifique et Technique–Office International des Epizooties*, 24(3):857–868, 2005.

G. Cringoli, L. Rinaldi, V. Veneziano, and V. Musella. Disease mapping and risk assessment in veterinary parasitology: some case studies. *Parassitologia*, 47(1):9–25, 2005.

S. de La Rocque, S.J.F. Michel, J. Bouyer, G. De Wispelaere, and D. Cuisance. Geographical information systems in parasitology: a review of potential aplications using the example of animal trypanosomosis in West Africa. *Parassitologia*, 47(1):97–104, 2005.

E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

A.E. Dooley, W.J. Parker, and H.T. Blair. Modelling of transport costs and logistics for on-farm milk segregation in New Zealand dairying. *Computers and Electronics in Agriculture*, 48:75–91, 2005.

L. Fu, D. Sun, and L.R. Rilett. Heuristic shortest path algorithms for transportation applications: State of the art. *Computers and Operations Research*, 33:3324–3343, 2006.

K. Ganesh and T.T. Narendran. CLOSE: a heuristic to solve a precedence-constrained travelling salesman problem with delivery and pickup. *International Journal Services and Operations Management*, 1(4):320–342, 2005.

M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, New York, 1979.

E.P. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

G. Hendrickx, J. Biesemans, and R. De Deken. The use of GIS in veterinary parasitology. In P.A. Durr and G.A.C. Gatrell, editors, *GIS and Spatial Analysis in Veterinary Science*, pages 145–176. CABI Publishing, Wallinford, UK, 2004.

R.R. Kao. The impact of local heterogeneity on alternative control strategies for foot-and-mouth disease. *Proceedings of the Royal Society London Part B*, 270:2557–2564, 2003.

S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

K. Kroschewski, M. Kramer, A. Micklich, C. Staubach, R. Carmanss, and F.J. Conraths. Animal disease outbreak control: the use of crisis management tools. *Revue Scientifique et Technique–Office International des Epizooties*, 25 (1):211–221, 2006.

M.P.M. Meuwissen, S.H. Horts, R.B.M. Huirne, and A.A. Dijkhuizen. A model to estimate the financial consequences of classical swine fever outbreaks: principles and outcomes. *Preventive Veterinary Medicine*, 42:249–270, 1999.

A. Nikolakopoulos and H. Sarimveis. A threshold accepting heuristic with intense local search for the solution of special instances of the travelling salesman problem. *European Journal of Operational Research*, 177:1911–1929, 2007.

J. Pacheco and R. Martí. Tabu search for a multi-objective routing problem. *Journal of the Operational Research Society*, 57:29–37, 2006.

E. Pinzon, J.M. Wilson, and C.J. Tucker. Climate-based health monitoring systems for eco-climatic conditions associated with infectious diseases. *Bulletin de la Societe de Pathologie Exotique*, 98(3):239–243, 2005.

D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers and Operations Research*, 34:2403–2435, 2007.

C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers and Operations Research*, 31:1985–2002, 2004.

A. Rizzoli, R. Rosa, B. Mantellig, E. Pecchioli, H. Haufe, V. Tagliapietra, T. Beninati, M. Neteler, and C. Genchi. Ixodes ricinus, transmitted diseases and reservoirs. *Parassitologia*, 46(1-2):119–122, 2004.

S. Ropke and D. Pisinger. An adaptive large neighbourhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472, 2006.

M. Sigurd, D. Pisinger, and M. Sig. Scheduling transportation of live animals to avoid the spread of diseases. *Transportation Science*, 38(2):197–209, 2004.

R. Tavakkoli-Moghaddam, N. Safaei, and Y. Gholipour. A hybrid simulated annealing for capacitated vehicle routing problems with the independent route length. *Applied Mathematics and Computation*, 176:445–454, 2006.

R. Tavakkoli-Moghaddam, N. Safaei, M.M.O. Kah, and M. Rabbani. A new capacitated vehicle routing problem with split service for minimizing fleet cost by simulated annealing. *Journal of the Franklin Institute*, 344:406–425, 2007.

List of Tables

Table 1: Overview of the number of farms to visit for each veterinarian determined on the basis of a real-life dataset.

Veterinarian	# farms	Veterinarian	# farms	Veterinarian	# farms
1	4	11	4	21	2
2	55	12	2	22	9
3	7	13	14	23	14
4	5	14	2	24	11
5	2	15	7	25	2
6	1	16	1	26	1
7	2	17	7	27	1
8	2	18	2	28	1
9	14	19	6	29	1
10	11	20	1	30	3

Table 2: Radii of the zones of higher risk as imposed for clasical swine fever and penalties (distance- and time-based) for entry in these zones. For conversion between distance and time, a vehicle velocity of 50 km/h was assumed.

Zone	Radius (km)	Penalty (km)		Penalty (h)	
		Entry	Disinfection	Entry	Disinfection
Quarantine	3	10	25	0.2	0.5
Surveillance	10	32	–	0.6	–

Table 3: Schedule for the collection of milk without precedence constraints. The zones in which farms are located are also indicated.

Trip 1		Trip 2		Trip 3	
Node	Zone	Node	Zone	Node	Zone
33849	quarantine	34040	surveillance	35187	free
33851	quarantine	33807	surveillance	23078	free
34056	quarantine	35959	free	34543	free
35103	surveillance	33795	free	35782	free
33853	surveillance	30934	free	33794	free
33847	surveillance	25120	free	34538	free
35438	surveillance	24713	free	35933	quarantine
32956	surveillance	35512	free	35114	quarantine
		34855	free		
Trip 4		**Trip 5**		**Trip 6**	
Node	Zone	Node	Zone	Node	Zone
33806	surveillance	33816	surveillance	15426	surveillance
13957	surveillance	13364	free	11765	surveillance
33829	quarantine	15976	surveillance	34808	surveillance
35839	quarantine	33827	surveillance	33843	quarantine
36001	quarantine	33835	quarantine	35235	surveillance
23801	quarantine	33834	quarantine	33959	free
23785	quarantine	34887	quarantine	33845	free
		35704	surveillance	35272	
		35305	surveillance		

Table 4: Schedule for the collection of milk in case of disease outbreaks. The zones in which farms are located are also indicated.

Trip 1		Trip 2		Trip 3	
Node	Zone	Node	Zone	Node	Zone
32956	surveillance	35959	free	23078	free
35438	surveillance	30394	free	35187	free
33847	surveillance	33795	free	33794	free
33853	surveillance	25120	free	35782	free
35103	surveillance	24713	free	34543	surveillance
34056	quarantine	35512	free	34538	surveillance
33851	quarantine	34855	free	35933	quarantine
33849	quarantine	33807	surveillance	35114	quarantine
		34040	surveillance		
Trip 4		**Trip 5**		**Trip 6**	
Node	Zone	Node	Zone	Node	Zone
33806	surveillance	13364	free	35272	free
13957	surveillance	33816	surveillance	33845	free
23801	quarantine	15976	surveillance	33959	free
23785	quarantine	33827	surveillance	35235	surveillance
36001	quarantine	35704	surveillance	34808	surveillance
33829	quarantine	35305	surveillance	11765	surveillance
35839	quarantine	34887	quarantine	15426	surveillance
		33834	quarantine	33843	quarantine
		33835	quarantine		

Table 5: Schedule broken into trips in case of disease outbreaks with indication of the zone in which the farm is situated and the volume of milk (ℓ) to be collected.

Trip 1			Trip 2			Trip 3		
Node	Zone	Vol. (ℓ)	Node	Zone	Vol. (ℓ)	Node	Zone	Vol. (ℓ)
35272	free	2551	35187	free	1850	33816	surveillance	1804
33845	free	2259	25120	free	2264	15976	surveillance	2251
33959	free	2137	24713	free	1812	33827	surveillance	2106
35959	free	1861	35512	free	2515	11765	surveillance	1978
30934	free	1844	34855	free	2689	34808	surveillance	1876
33795	free	1990	13364	free	2240	35704	surveillance	2396
35782	free	2534	34040	surveillance	2440	35305	surveillance	2531
33794	free	2075	33807	surveillance	2224	15426	surveillance	1913
23078	free	2606	35235	surveillance	2354			
Vol. (ℓ)		19857	Vol. (ℓ)		18034	Vol. (ℓ)		19209

Trip 4			Trip 5			Trip 6		
Node	Zone	Vol. (ℓ)	Node	Zone	Vol. (ℓ)	Node	Zone	Vol. (ℓ)
13957	surveillance	2630	32956	surveillance	2688	33843	quarantine	1864
34543	surveillance	2458	35438	surveillance	1806	35839	quarantine	2286
34538	surveillance	2111	33847	surveillance	1911	33829	quarantine	2244
33806	surveillance	2155	33853	surveillance	2486	36001	quarantine	2535
			35103	surveillance	1886	23801	quarantine	2493
			34056	quarantine	2079	23785	quarantine	2296
			33851	quarantine	2352	33834	quarantine	1804
			33849	quarantine	1926	33835	quarantine	1810
						34887	quarantine	1812
Vol. (ℓ)		9354	Vol. (ℓ)		17134	Vol. (ℓ)		19144

Trip 7		
Node	Zone	Vol. (ℓ)
35933	quarantine	2446
35114	quarantine	2063
Vol. (ℓ)		2309

Table 6: Cost of trips if a new schedule has been identified, following the precedence constraints and a maximum load capacity of 20000 ℓ and a maximum trip duration of 12 h and 5 h respectively.

max. 12 h		max. 5 h	
Cost (h)	Volume (ℓ)	Cost (h)	Volume (ℓ)
2.996	18467	2.996	18467
2.605	19424	2.605	19424
2.669	9354	4.695	19209
4.695	19209	1.986	10777
2.525	17134	3.56	13863
5.156	18856	4.234	17280
2.263	4509	3.632	8221

List of Figures

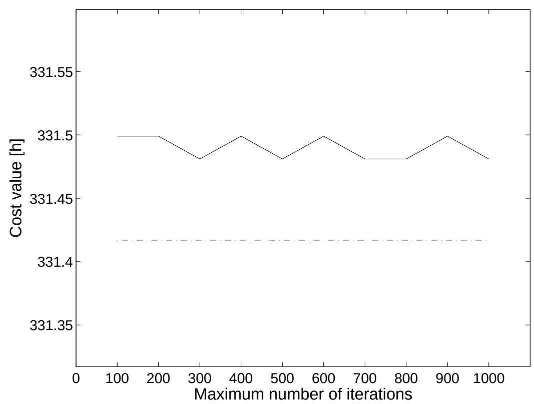


Figure 1: Minimal (dashed dotted line) and maximal (full line) costs out of 30 repetitions for schedules identified with TA for a veterinarian with 55 farms to visit.

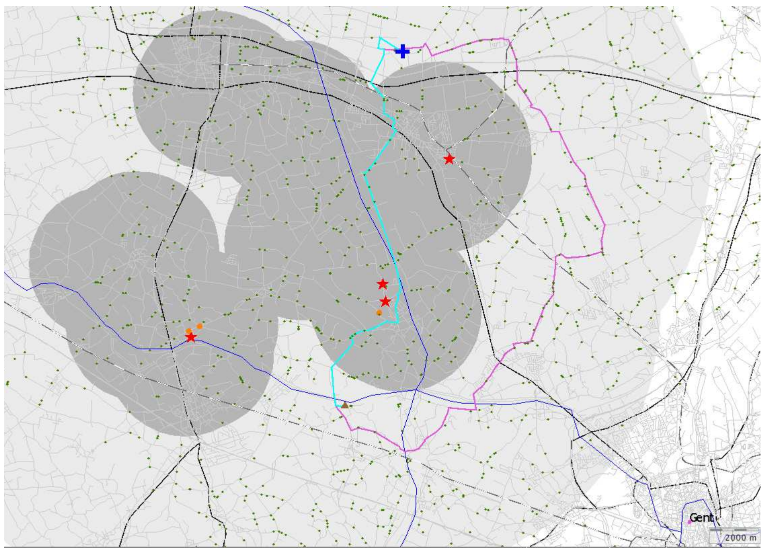
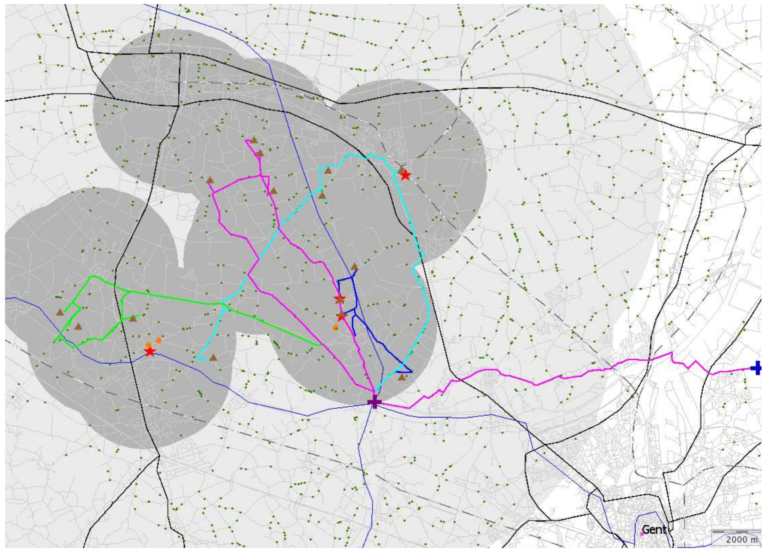
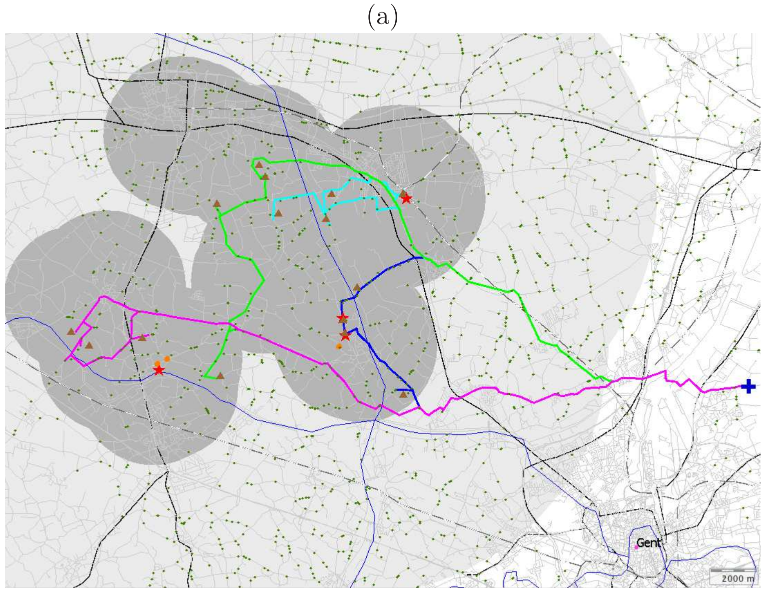


Figure 2: Path for a veterinarian based on the shortest path calculation if no penalty (blue) and a penalty of 10 km (magenta) is added for entering the quarantine zone. The vet's office is marked by a blue cross, the farm to visit by a brown triangle, infected farms by a red star, suspected farms by an orange dot and cleared farms by a green dot. The quarantine zone is marked in dark grey and the surveillance zone in light grey.

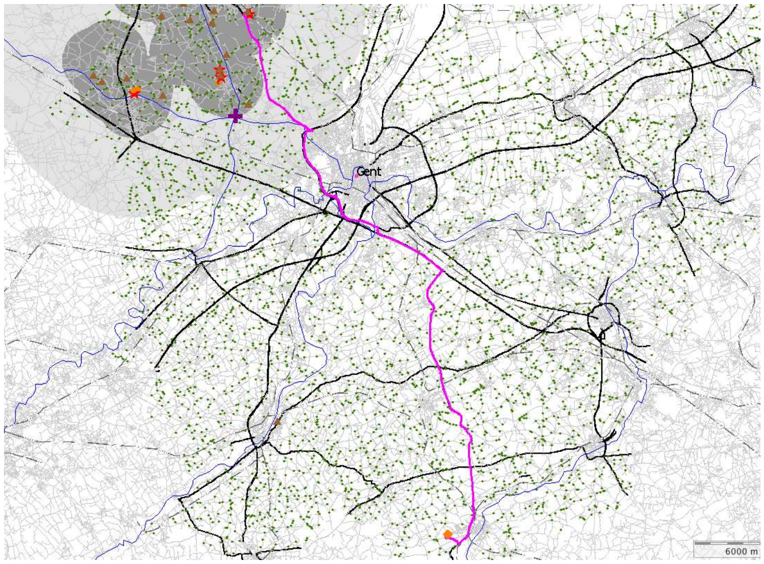


(a)

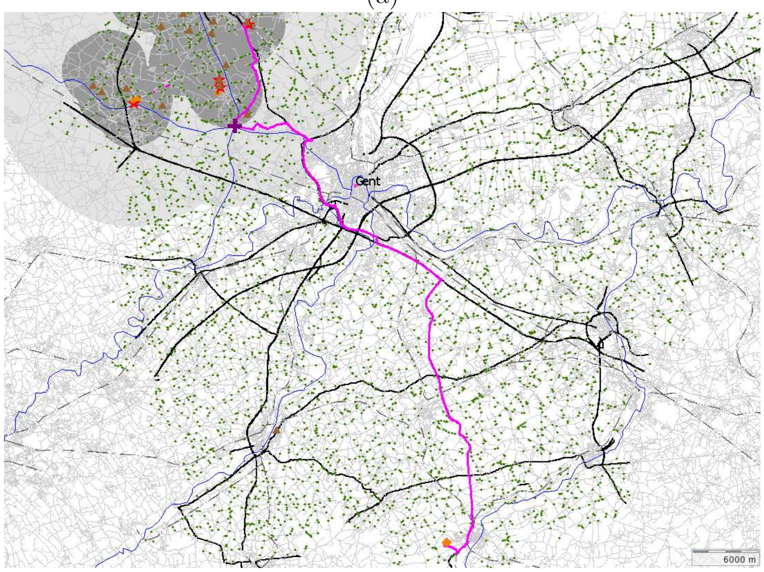


(b)

Figure 3: Path for a veterinarian based on the shortest path calculation for fixed (a) or mobile disinfection (b) units. The vet's office is marked by a blue cross, the farms to visit by a brown triangle, infected farms by a red star, suspected farms by an orange dot, cleared farms by a green dot. The quarantine zone is marked in dark grey and the surveillance zone in light grey. The fixed unit is indicated by a magenta cross. Paths for visits of the first to the fourth day are given in dark blue, cyan, green and magenta, respectively.



(a)



(b)

Figure 4: Path (magenta) for the transportation of cadavers if the vehicle is disinfected at the farm (a) or at a fixed disinfection location (b). The destruction company is marked by an orange diamond, the farm to visit by a brown triangle, infected farms by a red star, suspected farms by an orange dot and cleared farms by a green dot. The quarantine zone is marked in dark grey and the surveillance zone in light grey. The fixed disinfection unit is indicated by a magenta cross and is also indicated in (a).

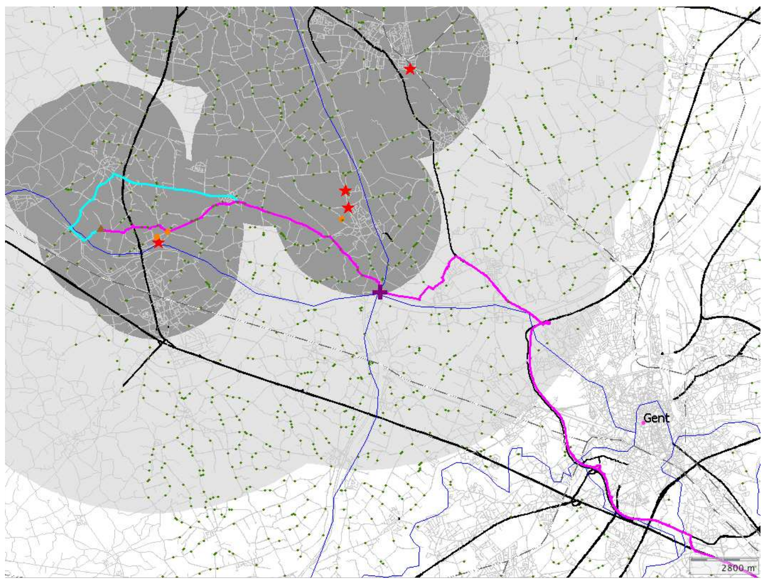


Figure 5: Path for the transportation of cadavers if a penalty is added for passing near not-infected farms (cyan) and if no penalty is added (magenta). The farm to visit is marked by a brown triangle, infected farms by a red star, suspected farms by an orange dot and cleared farms by a green dot. The quarantine zone is marked in dark grey and the surveillance zone in light grey. The fixed disinfection unit is indicated by a magenta cross.